

N86 - 30374

Special note to printer on MS "Integral Processing in beyond-Hartree-Fock calculations, by Peter R. Taylor"

The three "figures" should be typeset by the printer, as they appear. In particular, the indentation structure should be preserved exactly, as this is designed to illustrate computer program loops. The typefaces used should be somewhat smaller than those used for the text.

Integral processing in beyond-Hartree-Fock calculations

Peter R. Taylor
ELORET Institute*
Sunnyvale, CA 94087

Abstract

The increasing rate at which improvements in processing capacity outstrip improvements in input/output performance of large computers has led to recent attempts to bypass generation of a disk-based integral file. The "direct" SCF method of Almlöf and co-workers represents a very successful implementation of this approach. The present work is concerned with the extension of this general approach to CI and MCSCF calculations. After a discussion of the particular types of MO integrals for which — at least for most current generation machines — disk-based storage seems unavoidable, it is shown how all the necessary integrals can be obtained as matrix elements of Coulomb and exchange operators that can be calculated using a direct approach. Computational implementations of such a scheme are discussed.

* Mailing address: NASA Ames Research Center, Moffett Field, CA 94035

I. Introduction

One of the most interesting recent developments in computational quantum chemistry is the "direct SCF" approach of Almlöf, Faegri and Korsell (AFK) [1]. Recognizing that in some circumstances it may not be feasible to generate a disk file of two-electron integrals (or supermatrix elements) to be used repeatedly in subsequent SCF iterations, AFK suggested that the two-electron integrals be recalculated in each SCF iteration. That is, the Fock matrix contributions from each batch of integrals are computed and then each batch is discarded. There are two distinct sets of circumstances where this strategy should prove advantageous. Where computations are performed using an "in-house" minicomputer it will often be the case that the available disk storage is inadequate for large basis sets (150 - 200 CGTOs, say), or that the performance of the input/output (IO) system is too low for the integral file to be processed in an acceptable real time. Alternatively, where computations are being done on a supercomputer (or even a large conventional mainframe computer), the available disk system capacity and/or performance may not be adequate for the size of basis set (300 or more CGTOs) for which the integral generation time would be acceptable. (Even the arrival of large primary memories, such as the 268 million words available on the CRAY 2, does not provide a complete solution to the problem of storing the integrals). The direct SCF method has proved its worth in both sets of circumstances: basis sets of over 300 CGTOs having been handled on NORD 500 and VAX 11/780 minicomputers and over 500 CGTOs on an IBM 3033 [1]. Of course, AFK's implementation of the direct SCF method incorporates a number of factors designed to improve overall performance. The full symmetry of the nuclear framework is used to minimize the number of distinct two-electron integrals which *might* have to be calculated, while density matrix pre-screening techniques are used to avoid calculation of integrals which contribute negligibly to the Fock matrix [1,2].

While it is very desirable to have a method of this type, there are, of course, many chemical problems for which correlation effects play an important role. Consequently, it seems appropriate to explore schemes whereby a similar general approach — that is, recalculating integrals when they are required — could be taken for CI and MCSCF methods. This work presents an approach in which it is assumed that some integrals are required so frequently that it would be inefficient to recompute

them repeatedly, while other integrals can be recomputed as required. Clearly, there is an operational difference between this approach and the philosophy behind direct SCF, as in the latter it is assumed that all integrals are in the same class as far as frequency of use is concerned.*

II. MO Integrals in Direct MR-CI(SD).

The various types of MO integrals appearing in single and double excitation direct CI calculations have been discussed in detail by Siegbahn [3], Ahlrichs [4] and Saunders and van Lenthe [5]. These treatments cover not only the cases of one or several reference configurations, but also the case where the reference configuration is "internally contracted" [4,6-8]. It is clear from these treatments that it is desirable to have the integrals $[ij|kl]$, $[ij|ka]$, $[ij|ab]$ and $[ia|jb]$ (where i, j, \dots denote MOs occupied in at least one reference configuration, and a, b, \dots the remaining MOs; charge density notation has been used for the integrals) available in the MO basis: these integrals contribute to many different terms in different ways. If *Coulomb operator matrices* \mathbf{J}^{ij} and *exchange operator matrices* $\mathbf{K}^{ij\pm}$ are defined via the matrix elements

$$(p|J^{ij}|q) = [ij|pq] \quad (1)$$

$$(p|K^{ij+}|q) = [ip|jq] + [iq|jp] \quad (2)$$

$$(p|K^{ij-}|q) = [ip|jq] - [iq|jp] \quad (3)$$

for p, q, \dots arbitrary MOs, the above required integrals are all included in \mathbf{J}^{ij} , \mathbf{K}^{ij+} and \mathbf{K}^{ij-} , $i \geq j$.

* A note on terminology may be appropriate here. The expression "direct CI" has an accepted and widely understood meaning: it refers to a CI calculation in which the Hamiltonian matrix is never computed explicitly and stored. It is not unreasonable to use the term "direct SCF" for an SCF calculations in which the AO integral list is not stored. The expression "direct MCSCF" is closer in meaning to direct CI: the Hessian is not computed explicitly. It is difficult to combine these meanings to cover the sort of method proposed in this work, and while this author has previously used the term "direct direct CI" this is both ugly and confusing. No convenient alternative readily presents itself, however.

The remaining possible integral types, $[ia|bc]$ and $[ab|cd]$, are not required in the MO basis for direct CI calculations [4,5]. Their contribution to the residual vector $\sigma = \mathbf{H}c$ can be written in terms of AO integrals for a suitable renormalization of c [4,5,8]. This approach is discussed further in section V below.

It appears, therefore, that if a direct CI scheme is used for optimization of an MR-CI(SD) wave function, the integrals which must be computed (and stored) in the MO basis are just the \mathbf{J}^{ij} and $\mathbf{K}^{ij\pm}$ for *correlated* MOs $i \geq j$. It should be noted that while the term “CI” has been used here, all of the above remarks apply also to methods based on the coupled-pair many-electron theory of Cizek [9]. This includes both coupled-cluster methods and approximate CEPA-type schemes [10-13].

III. MO integrals in MCSCF calculations.

The question of which integral types need to be transformed into the MO basis has been investigated in some detail by Almlöf and Taylor [2]. Their conclusion is that it is generally necessary to have matrix elements of \mathbf{J}^{tu} and $\mathbf{K}^{tu\pm}$ in the MO basis: here t and u denote partially occupied (*active*) MOs in the MCSCF wave function. The availability of operators \mathbf{J}^{ij} , $\mathbf{K}^{ij\pm}$, \mathbf{J}^{it} and $\mathbf{K}^{it\pm}$ (here i, j, \dots denote doubly occupied (*inactive*) MOs) in the MO basis allows a very simple formulation of the MCSCF orbital optimization problem (see e.g. refs 14 and 15), but in a “direct” MCSCF formulation [14] it is always possible to rewrite the contributions of these operators in terms of the AO integrals [2,16]. Elements of \mathbf{J}^{tu} and $\mathbf{K}^{tu\pm}$ are needed for the CI step (or, in a full second-order treatment, the CI sub-block of the Hessian and the CI gradient term) and for some CI-orbital rotation coupling terms, and most of these contributions are awkward to reformulate in terms of AO integrals. In this way, each cycle of the MCSCF optimization requires construction of \mathbf{J}^{tu} and $\mathbf{K}^{tu\pm}$ once, followed by contraction of a supermatrix with quantities similar to density matrices. This contraction must be performed in every micro-iteration through the MCSCF linear equation system if a full second-order optimization is performed — for first-order schemes [15,17,18] intermediate Fock-type operator matrices can be constructed with one such contraction step and then re-used within the given cycle. Full details are given in ref 18.

For a second-order MCSCF scheme with the minimum number of integrals stored on disk, therefore, it will be necessary to recompute the AO integrals in every

micro-iteration of every cycle. In individual cases it may be preferable to construct \mathbf{J}^{ij} , $\mathbf{K}^{ij\pm}$, \mathbf{J}^{it} and $\mathbf{K}^{it\pm}$ once in each cycle, and then to process all the integrals in the MO basis: this would depend on the balance between the transformation labour to obtain these operators (and how many there are) and the integral evaluation time. For large extended systems it may be that sparseness in the integral list combined with pre-screening of density matrices might make the completely direct MCSCF approach favourable. Dynamic adjustment of the number of micro-iterations used in a given cycle (solving the linear equations less accurately when far from overall convergence) will also improve performance. In any event, for the purposes of the present discussion it is clear that the problem of generating MO integrals for use in an MCSCF calculation is equivalent to that of a CI calculation: J and K^\pm operators over certain occupied MOs must be available.

IV. Construction of operator matrices.

Where the AO integral list is available, and disk capacity or performance is adequate, the most efficient route to the required \mathbf{J} and \mathbf{K}^\pm matrices is via a limited four-index transformation [4,5] (see also ref 19 and refs therein), performed as the four quarter-transformations

$$[i\nu|\lambda\sigma] = \sum_{\mu} [\mu\nu|\lambda\sigma] C_{\mu i} \quad (4a)$$

$$[ij|\lambda\sigma] = \sum_{\nu} [i\nu|\lambda\sigma] C_{\nu j} \quad (4b)$$

$$[ij|p\sigma] = \sum_{\lambda} [ij|\lambda\sigma] C_{\lambda p} \quad (4c)$$

$$[ij|pq] = \sum_{\sigma} [ij|p\sigma] C_{\sigma q} \quad (4d)$$

for the element J_{pq}^{ij} . Here μ , ν , λ and σ denote AOs and \mathbf{C} is the matrix of MO coefficients. The most time-consuming of the four steps is (4a), which behaves as nN^4 operations for n active or correlated MOs and N AOs; (4b – d) behave as n^2N^3 . Similar behaviour is obtained for calculation of $K_{pq}^{ij\pm}$ provided that the AO integrals are sorted differently before the transformation.

A less efficient (in terms of floating-point operations) procedure essentially involves combining the first two quarter-transformations into a single step, generating,

say,

$$(\mu|J^{ij}|\nu) = \sum_{\lambda} \sum_{\sigma} [\mu\nu|\lambda\sigma] C_{\lambda i} C_{\sigma j} \quad (5)$$

and then transforming μ and ν to the MO basis. Defining “density matrices” \mathbf{D}^{ij} via

$$D_{\mu\nu}^{ij} = C_{\mu i} C_{\nu j} \quad (6)$$

allows (5) to be viewed as contraction of integrals with a density matrix, analogous to Fock matrix construction in an SCF calculation. (5) behaves as $n^2 N^4$, that is, some n times worse than (4). However, a scalar implementation of (5) requires no sorting of the AO integrals, and there is no need to expand the integral list beyond the normal canonical indexing $\mu \geq \nu$, $\lambda \geq \sigma$ and $(\mu\nu) \geq (\lambda\sigma)$.

Consider now an approach in which AO integrals are computed, used in some transformation process and then discarded, without being written to disk and re-read. If the $n^2 N^4$ process defined by (5) is used, it will be possible to hold simultaneously $2L/N(N+1)$ operator arrays \mathbf{J} or \mathbf{K}^{\pm} in L words of memory. As there are some $\frac{3}{2}n^2$ operators *in toto* to be constructed, it will be necessary to generate the integrals $3n^2 N^2/4L$ times. For 200 AOs, 20 correlated or active MOs and 4 million words of memory some 3 passes would be required, however, a 50% increase in n or N results in a factor of 2 increase in the number of passes, as would a 50% reduction in the memory available. The n^2 and N^2 scaling in the number of passes is clearly a considerable disadvantage of the $n^2 N^4$ approach.

On the other hand, by defining a “test density” as

$$D_{\lambda\sigma}^{test} = \max_{[ij]} |C_{\lambda i} C_{\sigma j}|, \quad (7)$$

where the notation $[ij]$ denotes all MO pairs whose operators are being processed in the current pass, an effective pre-screening technique can be implemented to decide whether a particular $[\mu\nu|\lambda\sigma]$ need be calculated. (This process is readily extended to the case of calculating AO integrals in shells, as discussed below and in refs 1 and 2). Clearly, as n or N increases, the number of operators generated in each pass decreases. It may be expected that, in turn, the sparsity of \mathbf{D}^{test} will increase (certainly \mathbf{D}^{test} cannot become less sparse) which will decrease the number of AO integrals to be calculated in each pass. This phenomenon will tend to offset the effect of the n^2 and N^2 scaling discussed above, and will play an important role when localized MOs are used.

Completion of the transformation of the \mathbf{J}^{ij} , etc, is also simple in the case of the n^2N^4 approach. Each operator matrix, once constructed in the AO basis, can be transformed to the MO basis and then written to disk directly. No additional sorting is required and the final operator matrices are in exactly the form required for "matrix-formulated" direct CI [4,5,20]. Typical loop structures for constructing various operators are discussed in section VI below.

In an implementation of the nN^4 scheme different procedures must be followed for the J and K cases. For J operators, it is necessary to compute blocks of integrals $[\mu\nu|\lambda\sigma]$, for all $\mu \geq \nu$ and for as many $\lambda\sigma$ ($\lambda \geq \sigma$) pairs as will fit in L words of memory. It is then possible to carry out the first two quarter-transformations (4a, b) for *all* ij ($i \geq j$) pairs. The resultant $[ij|\lambda\sigma]$ must then be written to disk, so that once all of the $[ij|\lambda\sigma]$ are available they can be re-sorted to AO \mathbf{J} matrices for the final half-transformation. Note that in the AO integral generation it is not possible to restrict consideration to the case $(\mu\nu) \geq (\lambda\sigma)$ (the normal canonical ordering): effectively, the integrals must be computed twice. For K^\pm integral blocks $[\mu\nu|\lambda\sigma]$, with all $\mu\lambda$ and for as many $\nu\sigma$ ($\nu \geq \sigma$) as can be held in memory, are transformed to $[i\nu|j\sigma] \pm [i\sigma|j\nu]$ for all $i \geq j$. Again, these half-transformed integrals must be re-sorted for the final transformation. Clearly, this latter ordering of $[\mu\nu|\lambda\sigma]$ is different from the J case and the n^2N^4 scheme. Indeed, it not only differs from the conventional ordering used in integral programs, but it also involves some redundant recomputation of integrals because of the need to have all $\mu\lambda$ pairs, not just $\mu \geq \lambda$. Essentially, the AO integrals must be computed four times. There are thus not only disk and IO overheads associated with the nN^4 scheme, but also additional CPU costs occasioned by recomputation of integrals. It will depend on the individual case whether these additional overheads offset the much more favourable floating-point behaviour of the transformation step relative to the n^2N^4 scheme. It should be noted that the disk space (and IO required) behave as n^2N^2 , which is usually very much less than the N^4 requirements for the initial sorting of a disk-based integral list for a conventional transformation. A disadvantage of the suggested implementation of the nN^4 procedure is that it is not possible to make as much use of pre-screening as in the n^2N^4 case. This is because the first half-transformation is used to produce $[ij|\lambda\sigma]$ for all ij from $[\mu\nu|\lambda\sigma]$ for all $\mu\nu$: the effective "test density" analogous to (6) would involve all ij pairs and would thus be as dense as the worst possible case for the n^2N^4 scheme. It is quite conceivable that in some cases, such as large extended

organic systems, the n^2N^4 approach with its effective pre-screening would be the method of choice, while for relatively compact systems of heavier atoms, such as polynuclear transition metal complexes, the nN^4 approach would be preferable.

V. External Exchange Operators.

As was noted above, it has been pointed out by several authors [4,5,8] that the direct CI contribution of the MO integrals $[ab|ci]$ and $[ab|cd]$ can be evaluated in the AO basis using the operator matrices \mathbf{K}^P with elements

$$K_{ab}^P = (a|K^P|b) = \sum_{\mu} \sum_{\nu} (\mu|K^P|\nu) C_{\mu a} C_{\nu b}, \quad (8)$$

where

$$(\mu|K^P|\nu) = \sum_{\lambda} \sum_{\sigma} [\mu\lambda|\nu\sigma] V_{\lambda\sigma}^P, \quad (9)$$

with

$$V_{\lambda\sigma}^P = \sum_p \sum_q \tilde{c}_P^{pq} C_{\lambda p} C_{\sigma q} \quad (10)$$

The ‘‘CI coefficient’’ array \tilde{c} is obtained as follows. For doubly-excited CSFs which differ only in virtual MO occupation (i.e. all have the same virtual MO spin-coupling and the same $(N_e - 2)$ -electron occupied MO part P (for N_e electrons correlated)) the various CI coefficients c_P^{ab} are collected into the array \mathbf{c}_P which is then renormalized to give \bar{c}_P according to refs [4,5]. We then have

$$\tilde{c}_P^{pq} = \bar{c}_P^{pq} + \sum_Q \delta_{ip} \delta_{dq} c_Q^a B_{iabd}^{PQ} \quad (11)$$

where B_{iabd}^{PQ} is a two-electron coupling coefficient and c_Q^a is the CI coefficient of a singly-excited CSF.

Clearly, the construction of \mathbf{K}^P in the AO basis using (8 – 11) parallels the construction of the \mathbf{K}^{ij} operators via the n^2N^4 scheme outlined above in section IV. Indeed, by explicitly recognizing that the two virtual MOs can be either singlet or triplet coupled it is possible to proceed via $\mathbf{K}^{P\pm}$ operators obtained from sums and differences of integrals as in eqns (2) and (3). Pre-screening via a test density matrix can be used to reduce the number of AO integrals which must be calculated, offsetting in part the n^2N^4 dependence of the K^P generation. However, the external exchange operator construction must be performed in each CI iteration, which

(when the time taken to re-evaluate the integrals is included) is likely to lead to its dominating the timing for calculations with large basis sets.

It is also possible to consider an alternative scheme for computing the contribution from the external exchange operators which shares features with the nN^4 scheme for \mathbf{J}^{ij} and $\mathbf{K}^{ij\pm}$. It is possible to form arrays \mathbf{K}^{cd} according to

$$K_{\mu\nu}^{cd} = \sum_{\lambda} \sum_{\sigma} [\mu\lambda|\nu\sigma] C_{\lambda c} C_{\sigma d} \quad (12)$$

and then, without any intermediate IO, to combine these half-transformed integrals with CI-coefficients as

$$K_{\mu\nu}^P = \sum_c \sum_d K_{\mu\nu}^{cd} c_P^{cd}. \quad (13)$$

The $K_{\mu\nu}^P$ would be written out to disk for re-sorting. The strategy would be to hold all $\lambda\sigma$ values in memory (in (12)) for as many $\mu\nu$ values as possible. The floating-point behaviour of (12) (assuming that in practice it would be performed as two successive quarter-transformations) is $(N - n)N^4$, while that of (13) is essentially $n^2(N - n)^2N^2$. Of course, the same recomputation of integrals is required for (12) as for the nN^4 approach to construction of $\mathbf{K}^{ij\pm}$ matrices discussed in the previous section.

For the case of the “externally contracted” CI method of Siegbahn [7], integrals such as $[ac|bd]$ are used not simply to form K_{ab}^P but rather to form Δ_P where

$$\Delta_P = \sum_a \sum_b K_{ab}^P c_P^{ab} \quad (14)$$

Here c_P^{ab} is a CI coefficient in a wave function obtained in the lowest order of perturbation theory. Δ_P need be constructed only once during the contracted CI calculation, and thus there is a very considerable advantage over the normal CI methods, since these require recalculation of the external exchange contribution in each iteration.

VI. Treatment of symmetry

The direct SCF implementation of AFK benefits enormously from the exploitation of symmetry. This is used to reduce the number of distinct integrals which must be computed, and to reduce the dimensions of the various matrices which must

be processed. It is well known that the incorporation of symmetry considerably improves the efficiency of conventional 4-index transformation and CI programs, and it is certainly desirable to extend these improvements to the present approach to beyond-Hartree-Fock methods. This is not difficult, although there are several points worthy of note.

First, the operators J^{ij} and $K^{ij\pm}$ will not always transform according to the totally symmetric irreducible representation of the molecular point group, G . Thus

$$RJ^{ij}R^\dagger = \sum_{\Lambda} D_{K\Lambda}^{\alpha}(R)^* J^{ij}, \quad (15)$$

where $R \in G$, and $D_{K\Lambda}^{\alpha}(R)^*$ is an element of a representation matrix for α , which may not be an irreducible representation. By choosing appropriate combinations of ij and their partner MOs in degenerate irreducible representations, it is possible to restrict attention to the case of α irreducible. In (15), therefore, J^{ij} would represent a combination of J operators which transform according to row K of irreducible representation α . In an SCF calculation, the Fock and density matrices transform according to the case of α being the totally symmetric irreducible representation, and for this case a straightforward scheme for using a list of symmetry-distinct AO integrals to construct "skeleton" matrices which are later symmetrized to give the full result has been derived by Dupuis and King [21], based on earlier work by Dacre [22] and Elder [23]. The present author has extended the Dupuis and King scheme to the case of non-totally symmetric operators [24]. The only difficulty that arises in this extension is the need for full representation matrices (not merely characters) in the symmetrization of the skeleton matrices. These can be calculated from the characters of the group and a chain of subgroups by an ingenious method due to Hurley [25].

It is thus possible to use the technique of ref 24 to generate integrals over MOs from a list of symmetry-distinct AO integrals. Use of the n^2N^4 scheme (5) for the transformation step leads to very similar processing as in the SCF case, as for (5) there is no need to order the integrals. Fig 1 shows the loop structure of an integral routine designed to implement this scheme. The loop structure is greatly simplified: most codes would feature double loops over centres and then shells on those centres. Loops over shell components have not been shown explicitly. In the figures, the *stabilizer* [24] of a shell or centre is that subgroup of G under which the centre is invariant. Distinct integrals are generated in terms of double coset

representatives for various pairs of stabilizers: for full details the reader is referred to Davidson [26]. As far as the overall loop structure of Fig 1 is concerned there is essentially no change from the SCF case, for which the statements in the innermost loop would simply add or subtract appropriate Fock matrix contributions.

It is also possible to handle symmetry in J operator construction by the nN^4 scheme (4) straightforwardly and a possible loop structure is given in Fig 2. However, complications ensue for the K^\pm operators. This is because integral evaluation schemes are based on charge densities (products of basis functions) and determining the symmetry-distinct AO integral list is also based on charge densities. Such an approach naturally works for J operators, since what is required is a list of $[\mu\nu|\lambda\sigma]$ with $\mu\nu$ fixed and all $\lambda\sigma$, and this is simply all charge densities $\lambda\sigma$ for the single charge density $\mu\nu$. Symmetry-distinct integrals are obtained from $[\mu R\nu|T(\lambda S\sigma)]$, where R, S and T are operators from the point group: the range of operators giving distinct integrals is determined by the symmetry transformation properties of the points on which the AOs are centred. Again, it is simple to work in terms of unique charge distributions $\mu R\nu$ and $\lambda S\sigma$ and their transforms, and to form all $T(\lambda S\sigma)$ for a fixed $\mu R\nu$. For K^\pm operators, however, what is needed from the list $[\mu R\nu|T(\lambda S\sigma)]$ are terms with $\mu T\lambda$ fixed and all possible $R\nu TS\sigma$. Not only is this clearly not charge distribution based, but the range of T operators giving distinct integrals cannot be determined until $\mu, \nu, \lambda, \sigma, R$ and S are known. This complicates the loop structure of the integral program, and, since it is usually desirable to compute information about charge distributions in the outermost possible loop, it will be necessary either to compute this information in inner loops or to compute information about all possible charge distributions in the outer loops, performing redundant work since some of these distributions will turn out to be non-unique. A nN^4 scheme loop structure for K^\pm operators, incorporating symmetry, is given in Fig 3, and the problems associated with K^\pm operators can be clearly seen by comparing Fig 3 with Fig 2.

VII. Computational considerations

The need for repeated calculation of AO integrals, particularly in implementations of the n^2N^4 transformation procedure (5), suggests that a primary goal must be an efficient integral evaluations scheme. This problem has received considerable

attention in the last fifteen years [27-29], and a number of very efficient schemes have been devised. A key feature of these schemes is the use of shells of basis functions, a shell being defined by a set of contracted Gaussian functions of the same L value, located on the same centre, with the same exponents and contraction coefficients but differing in their angular behaviour. Integrals over four such shells — a “shell block” of integrals — share many common factors, and avoiding redundant recomputation of these factors results in a substantial increase in efficiency. Such use of shells rather than individual basis functions is implicit in the loop structures of Figs 1 - 3. The use of shells requires a modification of the pre-screening procedure: clearly, as long as one integral in a shell block is required it will be necessary to compute the entire block. It is therefore convenient to define test densities (7) for shells rather than basis functions. Thus

$$D_{MN}^{test} = \max_{[ij]} |C_{\mu i} C_{\nu j}|, \mu \in M, \nu \in N \quad (16)$$

for shells M and N .

Most AO integral evaluation schemes are rather readily vectorized [30]. Integral evaluation is also a task which is suited to parallel architectures [30]. For the rest of this section, therefore, we shall assume that the problem of efficient integral evaluation has been solved and concentrate on the processing of the AO integrals once they are available.

The nN^4 transformation (4) is vectorizable in terms of successive matrix multiplications in which the innermost loop is of order N . For vector processors such as the CRAY machines, multiplication of matrices of this order leads to performance close to the theoretical maximum. For computers that require greater vector lengths to achieve maximum performance it is possible to write (4) as a set of “vector = vector + scalar*vector” (SAXPY [31]) operations of length n^2 to N^2 or even n^2N to N^3 [2]. It is also possible to perform the first half-transformation (4a,b) efficiently on a parallel architecture, by generating and processing subsets of integrals (such as $[\mu\nu|\lambda\sigma]$, $\forall \lambda \geq \sigma$ and fixed $\mu \geq \nu$) on each processor. However, the re-ordering and subsequent processing of the half-transformed integrals will require considerable data movement between processors; and the overall efficiency will depend critically on the speed of inter-processor communication [32]. For machines with a large common memory or solid-state disk this will obviously be much less of a problem than for polytope architectures, such as hypercubes, with relatively slow

data paths between nodes.

The n^2N^4 scheme (5) is straightforward to vectorize (in terms of SAXPYs) on the number of operator matrices which can be held in memory simultaneously. The maximum possible value is $\frac{3}{2}n^2$, when all J , K^+ and K^- operators can be processed in one pass. For large basis sets the memory requirements would usually be prohibitive, and a subrange of operators would be processed in each pass. This may lead to vector lengths too short for efficient processing. This scheme is very easy to adapt to parallel architectures: each processor simply generates a subset of the J^{ij} , etc. although this requires each processor to generate all the AO integrals if inter-processor communication is to be avoided. Of course, for multi-processor architectures with common memory, such as the CRAY X-MP or CRAY 2 the latter problem does not arise.

It is clear that similar reasoning can be applied to the external exchange contribution discussed in section V. Indeed, some additional steps which arise in this case, such as (10) and (13), are also readily vectorized. It therefore seems that processing of integrals along the lines described here can be made very efficient on most current generation computing machinery.

Finally, it may be useful to give an example of the data storage and recalculation requirements in a large CI calculation using the schemes suggested here. We consider a calculation on the molecule $\text{Fe}(\text{CO})_5$, similar to the largest calculations reported by Lüthi and co-workers [33], but using a larger basis. Assuming that an $[8s6p4d1f]$ basis is used for Fe and a $[4s2p1d]$ basis for C and O, there will be 233 AOs (using spherical harmonics) and 39 occupied MOs at the Hartree-Fock level. If only the Fe $3d$ and $4s$ and ligand σ lone pair electrons are correlated there will be 9 MOs correlated, if the ligand π electrons are included there will be 19. We assume that 4 million words of central memory are available. For 9 MOs correlated there will be 126 J^{ij} and $K^{ij\pm}$ operators, and using the n^2N^4 scheme all could be computed in a single pass over the integrals, using (5). If density matrices (6) are formed in advance, the storage for operator matrices is halved and two passes over the integrals would be required. The final operator matrices would require less than one million words of disk space, assuming that C_{2v} symmetry is used. Use of the full D_{3h} symmetry would reduce this even further. If the nN^4 scheme (4) is used, one pass each for J and K operators would be required: this would be equivalent to recomputing the integrals about six times. Re-sorting of the half-transformed

operators could be done in memory. For 19 MOs correlated the number of passes for the nN^4 scheme would not change, however, the n^2N^4 scheme would require about six passes over the integrals using (5), or nine using (5) and (6). In either case some 4 million words of disk space would be needed for the final operators. For the n^2N^4 case these calculations would all vectorize with a vector length greater than 60, which would be very efficient on machines such as the CRAY 1 or CRAY X-MP.

In each iteration of the direct CI it is most efficient to generate the contribution from the external exchange operators first. For 9 MOs correlated there are 81 external exchange operators to be computed, these could be generated in two passes using (9). For 19 MOs there are 361 operators, these would require five passes. Using (12) one pass only would be required for either 9 or 19 MOs correlated, but again this is equivalent to computing the integrals four times. The completed exchange operators can be used as the first contributions to the vector σ , which would be of length about 350 000 words for 9 MOs correlated, assuming C_{2v} symmetry, or 3 000 000 for 19 MOs correlated. In the latter case it would be necessary to process the CI coefficients from disk if all of σ is to be held in memory. Calculations on this scale would hardly be possible using a "conventional" disk-based transformation and direct CI approach.

It is clear that the overall labour in such a calculation, while substantial, is not unreasonably large for a modern supercomputer, or even a large mainframe. It is also clear that if the only consideration is to minimize the number of times the AO integrals are recomputed there is little to choose between the nN^4 and n^2N^4 transformation schemes, at least for calculations of this size.

VII. Conclusions

The present work is an attempt to outline some novel prospects for large basis set electronic structure calculations that include electron correlation. In general, the various approaches suggested are well suited to modern computer architectures and share the overall philosophy of avoiding or minimizing the disk-based storage and retrieval of integrals. Only certain MO integrals need be stored: no storage of AO integrals is required and the method is thus a natural generalization of the direct SCF method of Almlöf and co-workers.

Acknowledgements

I would like to thank J. Almlöf, C. W. Bauschlicher, A. C. Hurley, L. G. M. Pettersson, B. O. Roos and P. E. M. Siegbahn for many helpful discussions.

References

- [1] J. Almlöf, K. Faegri jr and K. Korsell, *J. Comput. Chem.*, **3**, 385 (1982).
- [2] J. Almlöf and P. R. Taylor, in "Advanced Theories and Computational Approaches to the Electronic Structure of Molecules", ed C. E. Dykstra (Reidel, Dordrecht, 1984).
- [3] P. E. M. Siegbahn, *J. Chem. Phys.*, **72**, 1647 (1980).
- [4] R. Ahlrichs, in "Proceedings of the 5th European Seminar on Quantum Chemistry", ed T. H. van Duijnen and W. C. Nieuwpoort (Max Planck Institute, Garching, 1982).
- [5] V. R. Saunders and J. H. van Lenthe, *Mol. Phys.*, **48**, 923 (1983).
- [6] W. Meyer, in "Methods of Molecular Electronic Structure Theory", ed H. F. Schaefer (Plenum, New York, 1977).
- [7] P. E. M. Siegbahn, *Int. J. Quantum Chem.*, **18**, 1229 (1980).
- [8] H.-J. Werner and E. A. Reinsch, *J. Chem. Phys.*, **76**, 3144 (1982).
- [9] J. Cizek, *J. Chem. Phys.*, **45**, 4256 (1966).
- [10] A. C. Hurley, "Electron Correlation in Small Molecules" (Academic Press, London, 1976).
- [11] W. Kutzelnigg, in "Methods of Molecular Electronic Structure Theory", ed H. F. Schaefer (Plenum, New York, 1977).
- [12] R. J. Bartlett, *Ann. Rev. Phys. Chem.*, **32**, 359 (1981).
- [13] R. Ahlrichs, *Comput. Phys. Commun.*, **17**, 31 (1979).
- [14] J. Olsen, D. L. Yeager and P. Jørgensen, *Adv. Chem. Phys.*, **54**, 1 (1983).
- [15] B. O. Roos, in "Methods in Computational Molecular Physics", eds G. H. F. Diercksen and S. Wilson (Reidel, Dordrecht, 1983).
- [16] G. B. Bacskay, *Chem. Phys.*, **61**, 385 (1981).
- [17] B. O. Roos, P. R. Taylor and P. E. M. Siegbahn, *Chem. Phys.*, **48**, 157 (1980).
- [18] B. O. Roos, *Int. J. Quantum Chem. Symp.*, **14**, 175 (1980).
- [19] I. Shavitt, in "Methods of Molecular Electronic Structure Theory", ed H. F. Schaefer (Plenum, New York, 1977); C. W. Bauschlicher, in "Advanced Theories and Computational Approaches to the Electronic Structure of Molecules", ed C. E. Dykstra (Reidel, Dordrecht, 1984).
- [20] R. Ahlrichs and C. Zirz, in "Proceedings of the Molecular Physics and Quantum Chemistry Workshop", ed P. G. Burton (Wollongong, 1980).
- [21] M. Dupuis and H. F. King, *Int. J. Quantum Chem.*, **11**, 613 (1977).
- [22] P. D. Dacre, *Chem. Phys. Lett.*, **7**, 47 (1970).
- [23] M. Elder, *Int. J. Quantum Chem.*, **7**, 75 (1973).
- [24] P. R. Taylor, *Int. J. Quantum Chem.*, **27**, 89 (1985).
- [25] A. C. Hurley, *Chem. Phys. Lett.*, **107**, 155 (1984).

- [26] E. R. Davidson, J. Chem. Phys., **62**, 400 (1975).
- [27] M. Dupuis, J. Rys and H. F. King, J. Chem. Phys., **65**, 111 (1976).
- [28] L. E. McMurchie and E. R. Davidson, J. Comput. Phys., **26**, 218 (1978).
- [29] V. R. Saunders, in "Methods in Computational Molecular Physics", eds G. H. F. Diercksen and S. Wilson (Reidel, Dordrecht, 1983).
- [30] D. Hegarty, in "Advanced Theories and Computational Approaches to the Electronic Structure of Molecules", ed C. E. Dykstra (Reidel, Dordrecht, 1984).
- [31] C. Lawson, R. Hanson, D. Kincaid and F. Krogh, ACM Trans. Math. Software, **5**, 308 (1979).
- [32] M. E. Colvin, H. F. Schaefer and R. A. Whiteside, private communication.
- [33] H. P. Lüthi, P. E. M. Siegbahn and J. Almlöf, J. Phys. Chem., **89**, 2156 (1985).

Fig 1. Loop structure for $n^2 N^4$ operator matrix generation

```

Loop on subranges of  $ij$  such that all matrices fit in memory
  Loop on shells  $M$ , stabilizer is  $\mathcal{M}$ 
    Loop on shells  $N$  ( $\leq M$ ), stabilizer is  $\mathcal{N}$ 
      Define  $\mathbf{R}$  as generators for double cosets  $MGN \forall G \in \mathcal{G}$ 
      Loop on elements  $R$  of  $\mathbf{R}$  generating shells  $RN$ 
        Define  $\mathcal{U}$  as stabilizer of  $M.RN$ 
        Loop on shells  $\Lambda$  ( $\leq M$ ), stabilizer is  $\mathcal{L}$ 
          Loop on shells  $\Sigma$  ( $\leq \Lambda$ , unless  $\Lambda = M$ , when  $\Sigma \leq N$ ), stabilizer is  $\mathcal{S}$ 
            Define  $\mathbf{S}$  as generators for  $\mathcal{LGS} \forall G \in \mathcal{G}$ 
            Loop on elements  $S$  of  $\mathbf{S}$  generating shells  $S\Sigma$ 
              Define  $\mathcal{V}$  as stabilizer of  $\Lambda.S\Sigma$ 
              Define  $\mathbf{T}$  as generators for  $\mathcal{UGV} \forall G \in \mathcal{G}$ 
              Loop on elements  $T$  of  $\mathbf{T}$  generating  $T(\Lambda S\Sigma)$ 
                Compute  $[\mu R\nu | T(\lambda S\sigma)] \forall \mu \in M$ , etc
                Accumulate contributions into  $J_{\mu R\nu}^{ij}, J_{T\lambda TS\sigma}^{ij}$ 
                 $K_{R\nu TS\sigma}^{ij}$  or whichever skeleton operator
                matrices are being generated in this pass
              End loop on  $T$ 
            End loop on  $S$ 
          End loop on  $\Sigma$ 
        End loop on  $\Lambda$ 
      End loop on  $R$ 
    End loop on  $N$ 
  End loop on  $M$ 
  Symmetrize operator matrices, complete transformation
  and write operators from this subrange to disk
End loop on subranges of  $ij$ 

```

Fig 2. Loop structure for nN^4 J operator matrix generation

```

Loop on shells  $M$ , stabilizer is  $\mathcal{M}$ 
  Loop on shells  $N (\leq M)$ , stabilizer is  $\mathcal{N}$ 
    Define  $\mathbf{R}$  as generators for double cosets  $MGN \forall G \in \mathcal{G}$ 
    Loop on elements  $R$  of  $\mathbf{R}$ , generating shells  $RN$ 
      Define  $\mathcal{U}$  as stabilizer of  $M.RN$ 
      Loop on shells  $\Lambda$ , stabilizer is  $\mathcal{L}$ 
        Loop on shells  $\Sigma (\leq \Lambda)$ , stabilizer is  $\mathcal{S}$ 
          Define  $\mathbf{S}$  as generators for  $\mathcal{L}GS \forall G \in \mathcal{G}$ 
          Loop on elements  $S$  of  $\mathbf{S}$ , generating shells  $S\Sigma$ 
            Define  $\mathcal{V}$  as stabilizer of  $\Lambda.S\Sigma$ 
            Define  $\mathbf{T}$  as generators for  $\mathcal{U}GV \forall G \in \mathcal{G}$ 
            Loop on elements  $T$  of  $\mathbf{T}$ , generating  $T(\Lambda S\Sigma)$ 
              Compute  $[\mu R \nu | T(\lambda S \sigma)] \forall \mu \in M$ , etc
              stored in memory, indexed by  $\mu, \nu, \Lambda, \lambda, \Sigma, \sigma, S$  and  $T$ 
            End loop on  $T$ 
          End loop on  $S$ 
        End loop on  $\Sigma$ 
      End loop on  $\Lambda$ 
    Form skeleton  $J_{\mu R \nu}^{ij}$  for each  $ij, i \geq j$  and  $\mu \in M, \nu \in N$ 
    symmetrize and write to disk
  End loop on  $R$ 
End loop on  $N$ 
End loop on  $M$ 
(Read back, re-sort and transform — loop structure not given)

```

Fig 3. Loop structure for nN^4 K operator matrix generation

```

Loop on shells  $M$ , stabilizer is  $\mathcal{M}$ 
  Loop on shells  $N$  ( $\leq M$ ), stabilizer is  $\mathcal{N}$ 
    Define  $\mathbf{H}$  such that  $H\mathcal{N} \forall H \in \mathbf{H}$ , are distinct left cosets of  $\mathcal{N}$ 
    Loop on elements  $H$  of  $\mathbf{H}$ , generating shells  $H\mathcal{N}$ 
      Loop on shells  $\Lambda$ , stabilizer is  $\mathcal{L}$ 
        Define  $\mathbf{R}$  as generators for double cosets  $\mathcal{M}G\mathcal{L} \forall G \in \mathcal{G}$ 
        Loop on shells  $\Sigma$ , stabilizer is  $\mathcal{S}$ 
          Define  $\mathbf{S}$  as generators for  $\mathcal{N}G\mathcal{S} \forall G \in \mathcal{G}$ 
          Loop on elements  $R$  of  $\mathbf{R}$ , generating shells  $R\Lambda$ 
            Define  $\mathcal{U}$  as stabilizer of  $M.R\Lambda$ 
            Loop on elements  $S$  of  $\mathbf{S}$  generating shells  $S\Sigma$ 
              Define  $\mathcal{V}$  as stabilizer of  $N.S\Sigma$ 
              Define  $\mathbf{T}$  as generators for  $\mathcal{U}G\mathcal{V} \forall G \in \mathcal{G}$ 
              If  $H \in \mathbf{T}$  then
                Compute  $[\mu R \lambda | H(\nu S \sigma)] \forall \mu \in M$ , etc
                stored in memory, indexed by  $\mu, \nu, \Lambda, \lambda, \Sigma, \sigma, R, S$  and  $H$ 
              Endif
            End loop on  $S$ 
          End loop on  $R$ 
        End loop on  $\Sigma$ 
      End loop on  $\Lambda$ 
    Form skeleton  $K_{\mu H \nu}^{ij}$  for each  $ij, i \geq j$  and  $\mu \in M, \nu \in N$ 
    symmetrize and write to disk
  End loop on  $H$ 
End loop on  $N$ 
End loop on  $M$ 
(Read back, re-sort and transform — loop structure not given)

```